www.XtremePapers.com

UNIVERSITY of CAMBRIDGE
International Examinations

GCE Advanced Level Computing
Scheme of Work

Paper 4
Computing Project

Introduction

The project is a substantial piece of work requiring analysis and design over an extended period of time, which is organised, evaluated and presented in a report.

Candidates choose, in conjunction with their teacher, a well-defined user-driven problem of an appropriate size, which enables them to demonstrate their skills in Analysis, Design, Software Development, Testing, Implementation, Documentation and Evaluation.

Guidance on Selecting Projects

The selection of the problem for which a computerised system is to be designed and implemented is extremely important. It should be chosen by the candidate in consultation with the teacher, and should always involve a user, ideally a 'third party' user.

It is important to stress that the candidate should endeavour to produce a system which is non-trivial and which will solve a given problem sensibly within the constraints of resources available to the candidate.

Since the computing project seeks to assess the systems analysis section of the specification in a practical manner, candidates should not produce a system from their own limited knowledge of the requirements of the system. The 'third-party' user has to be someone who is willing to be involved in the project:

- in the analysis of the problem, where the user's requirements are obtained. This may take the form of a recorded interview with the candidate.
- at the software development, testing and implementation stages, where the user is involved in 'prototyping'.
- at the evaluation stage, where the user is involved in checking that the system is completed as specified and, leading on from this, is then willing to write a letter of acceptance of the system, including any criticisms of it.

In this way, candidates can be encouraged to look beyond school, or college, life into the businesses and companies in the community of the surrounding area. The emphasis is on analysing an existing system, and producing a computer-based solution to fit the needs of the user.

Projects should be chosen to demonstrate the integrative aspects of the work and should avoid needless repetition of the demonstration of a given skill. Each candidate must submit a report on their piece of work, supported by evidence of software development and testing. The solution may be implemented using one or more of the following:

- a programming language
- pre-written modules or toolkits
- applications software
- programmable packages

The teacher marks the projects according to the guidance on marking projects provided in the syllabus by CIE, after which moderation takes place according to CIE procedures.

Recommended Prior Knowledge

Candidates should have studied Paper One (Computer Systems, Communications and Software) and Paper Two (Structured Practical Tasks).

© UCLES 2006

Tutor Preparation Required to Start Paper 4

Decide on the programming languages and/or packages to be used, ensuring that candidates can fulfill all the syllabus requirements with the selected language or package. If a programming language is chosen then it could be the same as used for Paper One.

- ensure that the students have access to a word-processing package for report writing
- install the software the students will need so that it is accessible to all students
- provide instructions showing what the students have to do
- provide an outline of the project requirements and the marks awarded for each section
- prepare a list of suitable project ideas
- prepare a list of suitable types of end users
- prepare a bank of project reports for students to study

Important Note

Some centres may wish to deliver elements of this module alongside Paper Three (Systems, Software Mechanisms, Machine Architecture, Database Theory, Programming Paradigms and Integrated Information Systems). To help centres that wish to use this approach, the relevant session plans and performance criteria for the work on Papers One, Two and Three are mapped for each element listed in Paper Four.

© UCLES 2006

Scheme of Work

The classroom exercises are designed to help the student through the stages of project development from choosing a project to the final evaluation. They should be interspersed with project development sessions, where the students develop their individual projects. A suggested plan is set out in the Course Programme.

Session Plan 119 – Introduction

Assessment Objectives	Classroom Ideas	MS	MPC
Project Requirements	• choosing a project • stages in project development • documenting a project • planning project development • examiners' requirements	13 114	1.7.1 – 1.7.9 3.8.4

Classroom Exercises	Notes
Introduce the project work, using pre determined scenarios to introduce the idea of a 'third party' user rather than the teacher or the student. Also explain the stages at which third party involvement is required: • analysis • software development • evaluation Hints on finding a 'third party' user: • find a suitable problem to solve for another person • consider school/college, clubs and societies, small businesses, charities • ask family, friends, neighbours, employers, teachers etc. • look at manual systems, partially computerised systems, fully computerised systems Get students to brainstorm ideas for projects and users and discuss their suitability as a group. If possible have project reports from previous students available (both good and bad) for the students to look at. NB If students cannot find a 'third party' user of their own, it is better to persuade a colleague to fulfil this role rather that the teacher attempting to be both teacher and user.	An overview of all stages of the project should be included. Because it is valuable for students to be able to see an overview of the whole process. Definition, Investigation and Analysis: • definition - nature of the problem (3) • investigation and analysis (8) Design: • nature of solution (7) • intended benefits (2) • limits of the scope of the solution (2) Software development, testing & implementation: • development and testing (9) • implementation (6) • appropriateness of structure and exploitation of available facilities (3) Documentation: • technical documentation (6) • user guide (6) Evaluation: • discussion of the degree of success in meeting original objectives (3) • evaluate the users' response (3) • desirable extensions (2) These elements are taken from the marking scheme. The marks available for each element are shown in brackets.

Resources

- sample project ideas
- requirements for each section
- outline of report requirements

Key

MS = Mapping Session
MPC = Mapped Performance Criteria

Classroom Exercises
Prepare a document showing the requirements for each section and the marks awarded. Use this to introduce the idea of formal time planning for their project. When the students have chosen their project topic, get them to draw up a Gantt chart to use as a working document in planning their project. Perhaps show the students an outline template for their Gantt chart. Also encourage students to keep a project diary showing any major decisions made.
Introduce the idea of word-processing their report as the project develops. Provide an outline of the report structure, perhaps using a template. Stress the use of word-processing features to provide a professional report including the following: 1. Title and contents page 2. clear and logical organisation - set out in sections 3. continuous page numbering 4. information presented in continuous prose 5. good use of grammar, punctuation and spelling 6. use of a clear font style and size e.g. Times New Roman 10 pt or 12 pt 7. use of appropriate techniques e.g. headers and footers, consistent styles for headings, use of tables etc. 8. footers should contain the candidate's name and page number 9. inclusion of a bibliography

© UCLES 2006

Session Plan 120 – Definition, Investigation and Analysis

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.1**		Ask the students to bring in their project ideas and discuss with individuals the following:	13	1.7.1 1.7.3
	4.1.1 Define a problem	• define the nature of the problem to be solved		
	4.1.2 Investigate the current system	• identify methods by which to investigate the problem		
	4.1.3 Record findings	• record information/data and gather sample documents currently used (this could be set as a homework exercise)	113	1.7.4 1.7.5 3.8.1
	4.1.4 Analyse findings	• identify the current processes and current data structures		2.1.4
		• analyse the data and processes. For each process, work out the data required for input and describe the way that it is out put.		1.4.2
	4.1.5 Identify problems/ inefficiencies with current system	• specify inefficiencies and problems apparent from discussions with the user and the analysis work carried out		
	4.1.6 Specify Requirements	• derive the user and information requirements of the system	14	1.7.6
		• specify the required hardware and reasons for choice	1 2	1.1.1 1.1.2 1.2.1
		• specify the required software and reasons for choice	11	1.5.2 1.5.3
		• develop and document a clear requirement specification	15 17	1.8.1 1.8.2 1.8.3 1.10.2

Resources

- examples of good and poor problem definitions
- revision of different methods of fact finding
- revision of different methods of system documentation

© UCLES 2006

Classroom Exercises	Notes
Give the students a number of prepared problem definitions of varying quality and use these to discuss the characteristics of a good problem definition. The students set out their problem definition and word-process it. These problem definitions need to be checked for suitability by the teacher before the investigation and analysis proceeds. (4.1.1)	
Review the different methods of fact-finding and why they are used: • questionnaires • observation • structured interviews • also include a document search Stress the importance of using these with their user rather than the student relying on their own perceptions and ideas. Students should select appropriate methods of fact finding for their project, prepare appropriate documents for recording their findings and employ them with their user. Tell the students that they should be able to justify their choices. Stress the importance of keeping good records at this stage, including sample copies of documents used by the current system, to refer to at a later stage, as necessary. Students should use information to try and identify the data flows, data structures and the processing done by the current system. Also, encourage students to pinpoint inefficiencies and problems at this stage. This will help them identify possible needs for the development and maintenance of the system. **Hint** The need for further information may be identified at this stage – students should contact the 'third party' user if this is so before trying to proceed with incomplete information or guessing what is required. (4.1.2)	Discuss the features of good questionnaire design, such as unambiguous questions and attractive presentation.
Review the different methods of system documentation including: • structure diagrams • data flow diagrams (level two will include data stores and processes) • system flowcharts Students should establish the data requirements and produce a full conceptual data model from these and document any constraints and assumptions. Identify and document any library elements e.g. general input routines, calculating interest payments etc. Students should then use this information to set out a reasoned rationale for the system requirements including the following: • hardware and reasons for choice • software and reasons for choice • user requirements broken down into specific and measurable objectives (4.1.6)	Stress the importance of explaining the nature of the organisation and the methods currently in use. The report should include evidence of a high degree of user involvement, with detailed recording of the user requirements. Alternative approaches to the problem should be fully explained. This 'session' may take place over a number of weeks as the students develop their project ideas.

© UCLES 2006

Session Plan 121 – Documentation, Technical Manual

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.4**	4.4.1	• Development of a detailed technical manual	14 28	1.7.6 2.4.3

Classroom Exercises	Notes
Discuss the contents of technical manuals for given solutions. Encourage students to develop these alongside the project implementation. The contents should include at a minimum: For a program • operating environment (hardware, software) • data structures, variables defined by name, type and function • clear definition and identification of purpose of subroutines and functions • annotated code (try and use features available in the language chosen e.g. naming conventions, good layout and use of commenting) • instructions for modification • anything else that will help to maintain or update the system For a package • operating environment (hardware, software) • algorithm descriptions for each section of the solution • purpose and inter-relationship of modules within the system • data structures • annotated listings of macros • annotated design screens showing details of package-generated forms, reports, queries, buttons, cross tabulations etc. • instructions for modification • anything else that will help to maintain or update the system	Documentation for both a programmed solution and a package-based solution should be discussed. Some students may be using both - bear this in mind during the discussion. Technical Manual • annotated code/macros and/or annotated design screen views of other elements e.g. tables, forms, reports, QBE grids etc • data dictionary (including all elements developed) • instructions for modification • operating environment (hardware, software) Remind the students that the technical guide is a stand-alone document. They should think about the purpose of the document when they are writing it, which is *to make it easy for someone who does not know the system to maintain and upgrade the system*.

© UCLES 2006

Session Plan 122 – Documentation, User Guide

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.4**	4.4.2 User Guide	• development of a detailed user manual	27 118	2.4.2 3.10.3

Classroom Exercises	Notes
Discuss the contents of user manuals. Encourage students to develop these alongside the project implementation. **Hint:** Saving appropriate Input/Output screens (including error messages) to illustrate the user manual as the solution is developed and tested can save a lot of time. The user guide is built up as the program develops. It is a good idea to begin creating the manual at this stage, but students should be prepared to be flexible as the program may change as time develops. The advantages of using a word processor for the task are clear here. The user manual should be structured as a stand-alone document with contents or index, page numbers etc. and should include at a minimum: • step by step instructions for operating all aspects of the system including installation (use screen dumps of error messages as appropriate and test the instructions with the user to ensure that they are at an appropriate level for the target audience) • how to deal with any errors that may occur (use screen dumps of error messages as appropriate) • appropriate Help (in small meaningful easy to read sections), and messages within the software solution or in the form of a hypertext document (again these should be developed alongside the main solution) • backup procedures • glossary of terms used	User Manual includes: • step by step instructions for operating all aspects of the system • how to deal with any errors that may occur • appropriate Help and messages within the software solution, or in the form of a hypertext document • backup procedures • glossary of terms used

© UCLES 2006

Session Plan 123 – Design

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.2**	4.2.1 Output Design 4.2.2 Input Design	• design and document Input/Output	16 21 17	1.9.1 1.9.2 1.9.3 1.2.2 2.1.3 1.10.1
	4.2.3 Data Structures /Model	• design data structures necessary to solve the inefficiencies /problems indicated in the requirements specification • include full documentation of the data structures.	20 14 22 105	2.1.2 1.7.5 1.7.6 2.1.4 3.4.4
	4.2.4 Process Model	• design and document an algorithm /pseudo-code/top-down diagram or other form of process model	110 106	3.6.1 3.6.2 3.4.4

Classroom Exercises	Notes
Detailed models of the proposed interface (discuss these with the 'third party' user before implementation and show any changes made). Include: • report layouts • screen displays • other forms of output (for example, charts, sound, animation) • data capture forms (these could be online) • screen layouts Data structures that solve the inefficiencies identified in the requirements specification. Discuss: • normalisation • E-R models Include a process model and processing tasks, discussing where appropriate: • JSP • Algorithms Discuss how to identify benefits and limitations of the proposed system. Show students how to estimate the size of files required.	Emphasise the importance of uncluttered screens and clear instructions. Ensure that students address the issue of data validation, if necessary. Allow the students enough time to develop their designs. The design should include details of **all** aspects of the system; for example, data capture forms.

© UCLES 2006

Session Plan 124 – Software Development, Testing and Implementation

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.3**	4.3.1 Software development	• implement the proposed process model using a software package and/or programming language	108 3 24	3.5.2 1.3.1 1.3.2 2.2.2
		• develop the data structures of the design using the appropriate features of a software package and/or programming language	23	2.2.1
		• develop inputs/outputs using the features of a software package and/or a programming language		
	4.3.2 Test strategy/plan	• identify, develop and document a test strategy for the software solution	6 14 25	1.3.4 1.7.6 1.7.7 2.3.1
	4.3.3 Test data	• select suitable test data to carry out the test strategy		2.3.2 2.4.1
	4.3.4 Testing a software solution and planning for its implementation	• test the software solution, illustrating how the software solution evolves		
		• produce detailed output from the testing, cross-referencing as appropriate to the test plan		

Classroom Exercises

Review testing (look at Session Plans 25 and 26), the development of a test strategy, and the use of test plans and test data. Provide the students with an outline for a test plan to ensure all elements are fully tested and modifications are recorded.

Test plans are often best set out on tabular form clearly cross referenced to annotated output

Test No	Purpose	Data	Expected Results	Actual results	Remedial Action

Set students homework to develop one routine and the associated test plan. Use classroom discussion of the test plans to ensure that all types of testing are included and any modifications recorded. Allow students to test and modify this routine.

Set up peer testing of the finished routines to simulate user testing.

Notes

Encourage students to fully justify the choice of data.

Explain clearly the difference between valid and invalid data and demonstrate how extreme data can be either valid or invalid. Relate your discussion to normal, erroneous and extreme data.

Make sure the students can answer the question:
'If the system fails, when does it fail?'

© UCLES 2006

Session Plan 125 – Software Development, Further Testing including User Testing

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.3**	4.3.5 User Testing	• test the software solution with the user, providing documented evidence that the solution works • devise a strategy for its implementation		1.7.7 3.8.3 3.7.4

Classroom Exercises	Notes
Re-enforce and extend previous work done including: Expected types of test for a test strategy: • white box testing • black box testing • alpha testing with fellow students • beta testing involving the 'third party' user • acceptance testing with user • testing of user documentation Ensure software development takes place alongside this test strategy. (4.3.2) Discuss how to set up and document an implementation strategy and plan. Include the following: • user testing • user acceptance testing • obtaining feedback from user • installation of system • user training • system changeover **Hint:** Look at the guidance on marking projects to advise candidates on how to improve.	Include: • normal data • erroneous data • extreme data Include: • system testing • user testing with written feedback Discuss the implementation strategy, looking at the options: • parallel • phased • pilot • direct Emphasise to the students that they must be able to justify their choice of strategy. Keep a word processed diary of the testing and note any corrective action taken (4.3.4) Make sure that the students understand that they need to produce evidence of user testing. Their evidence could include: • user comments • photographs • screenshots • interview data

© UCLES 2006

Session Plan 126 – Evaluation

Assessment Objectives	Performance Criteria	Classroom Ideas	MS	MPC
• **4.5**	4.5.1 Evaluate results against the requirement specification	• evaluate the final system against the criteria described in the requirements specification	14 114	1.7.6 3.8.4
	4.5.2 Evaluate user testing	• evaluate the users' responses to testing the system		
	4.5.3 Identify the good and bad points of the final system	• identify the good and bad points of the final system highlighting any limitations and necessary extensions to the system, indicating how the extensions could be carried out		

Classroom Exercises	Notes
Provide an outline of the evaluation required and discuss how the students should develop it to match their solution. The outline should include:	
1. Comparison between the requirements specification and system developed, each criterion should be considered in the following way: - was it met? - if yes, how was it met? - if not, why not, and what limitations does this bring to the system?	1. look at each objective separately 2. emphasise to the students the need for a user-friendly solution 3. it is important that the user is actively involved in the evaluation
2. Evaluation of user testing with typical data: - feedback from user - changes made as a result of this feedback	Discuss how to obtain feedback from the user. For example, use questionnaires or interviews.
3. Use the above to summarise the good and bad points of the system: - identification of limitations - extensions needed and how they could be carried out	

© UCLES 2006